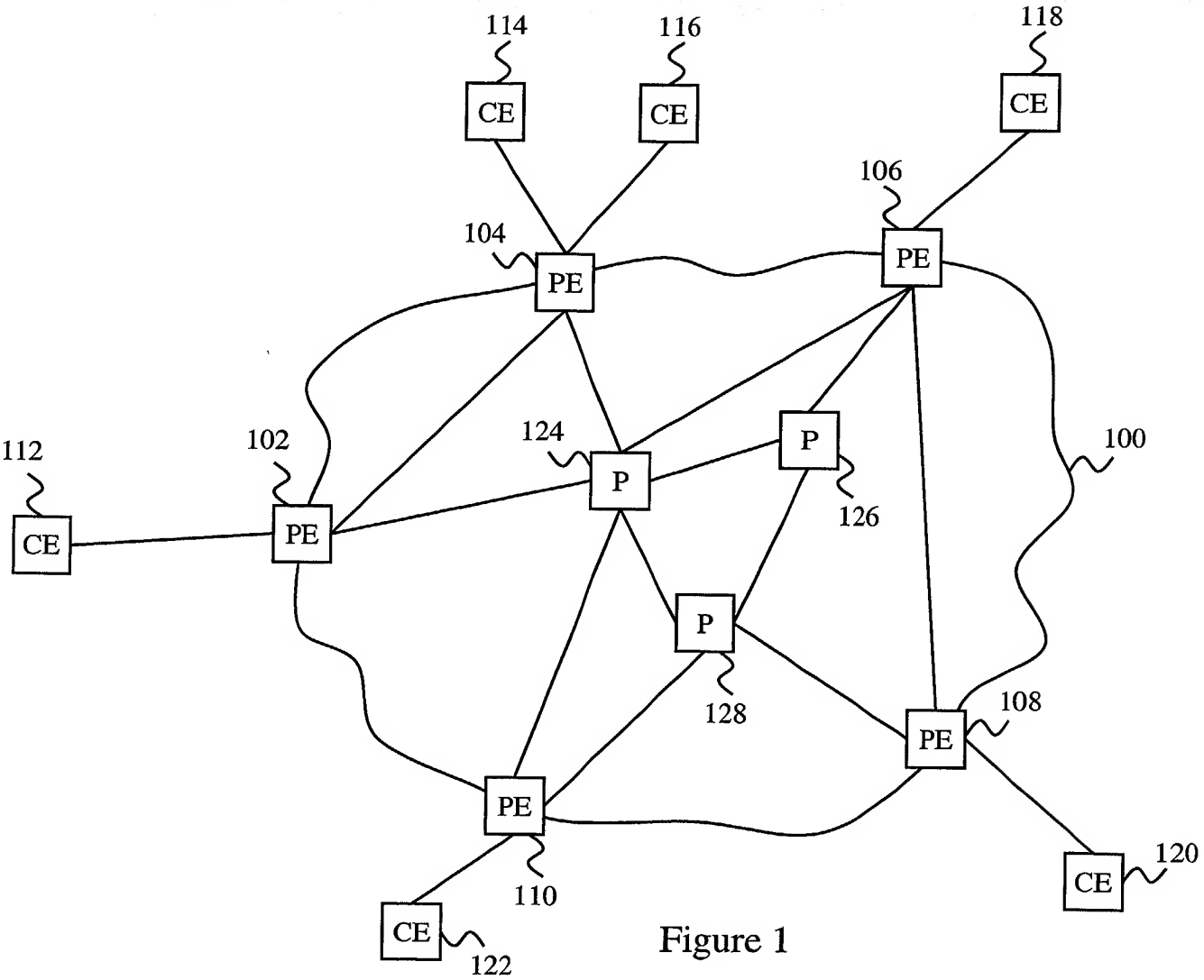
114
116
118
CE
CE
CE
106
104
PE
PE
112
102
124
P
P
100
CE
PE
126
P
128
108
PE
PE
110
120
CE
CE
122

Figure 1

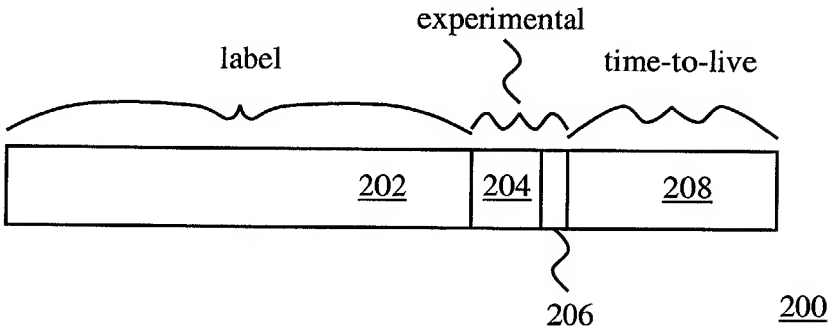

Figure 2

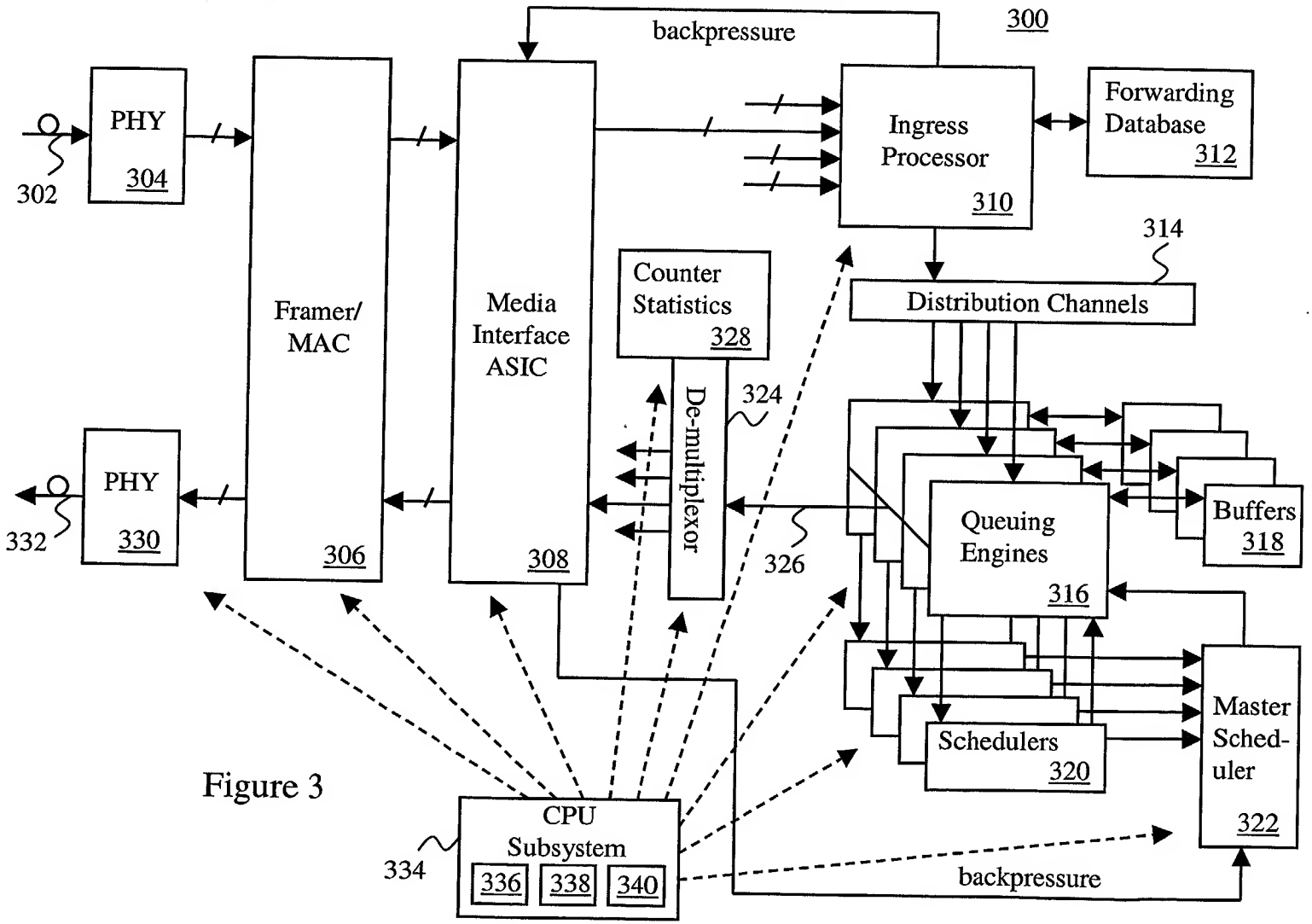
backpressure
300
PHY
304
302
Framer/
MAC
306
Media
Interface
ASIC
308
Ingress
Processor
310
Forwarding
Database
312
314
Distribution Channels
Counter
Statistics
328
De-multiplexor
324
326
PHY
330
332
Queuing
Engines
316
Buffers
318
Schedulers
320
Master
Sched-
uler
322
CPU
Subsystem
336
338
340
334
backpressure

Figure 3

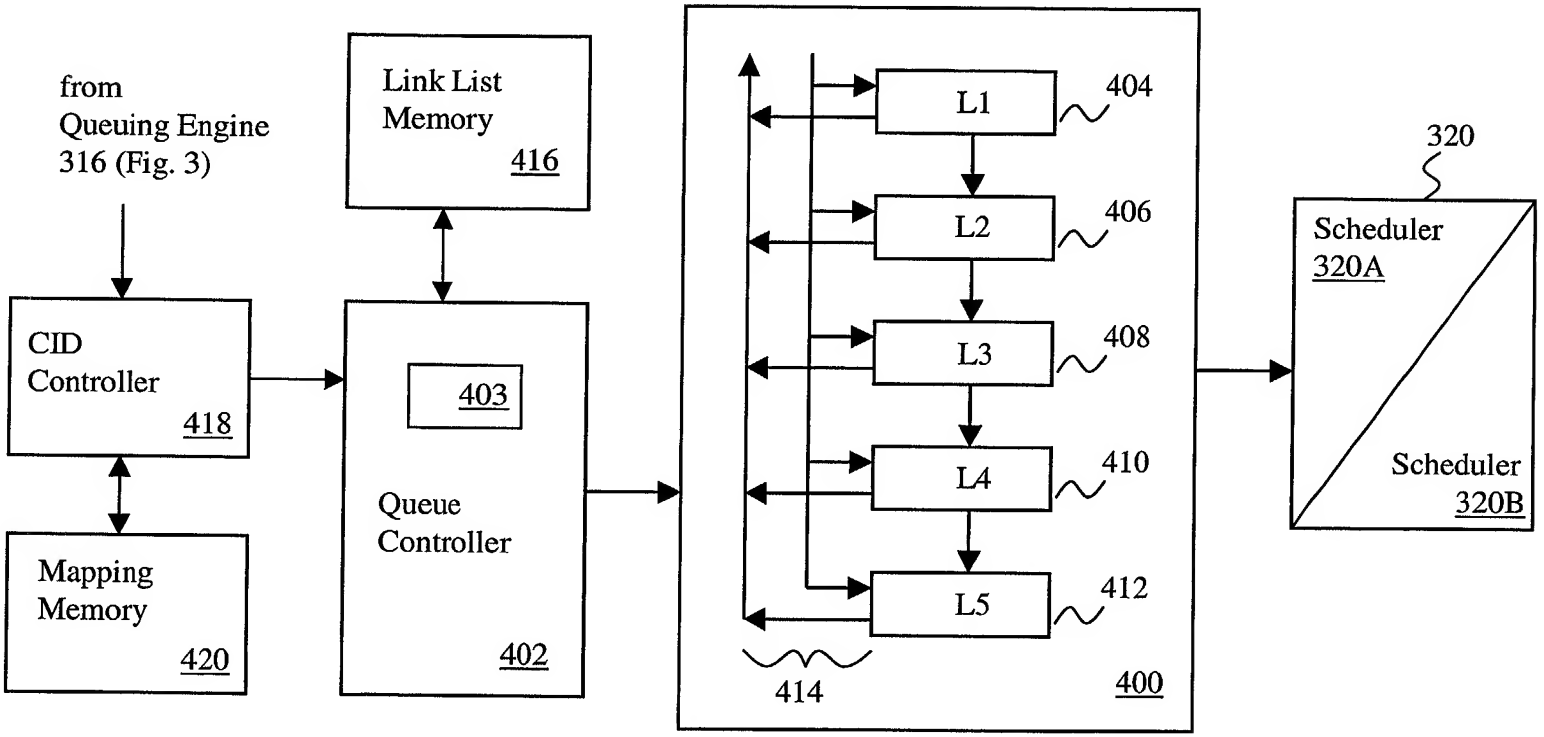
from
Queuing Engine
316 (Fig. 3)
Link List
Memory
416
CID
Controller
418
Mapping
Memory
420
403
Queue
Controller
402
L1
L2
L3
L4
L5
404
406
408
410
412
414
400
320
Scheduler
320A
Scheduler
320B

Figure 4

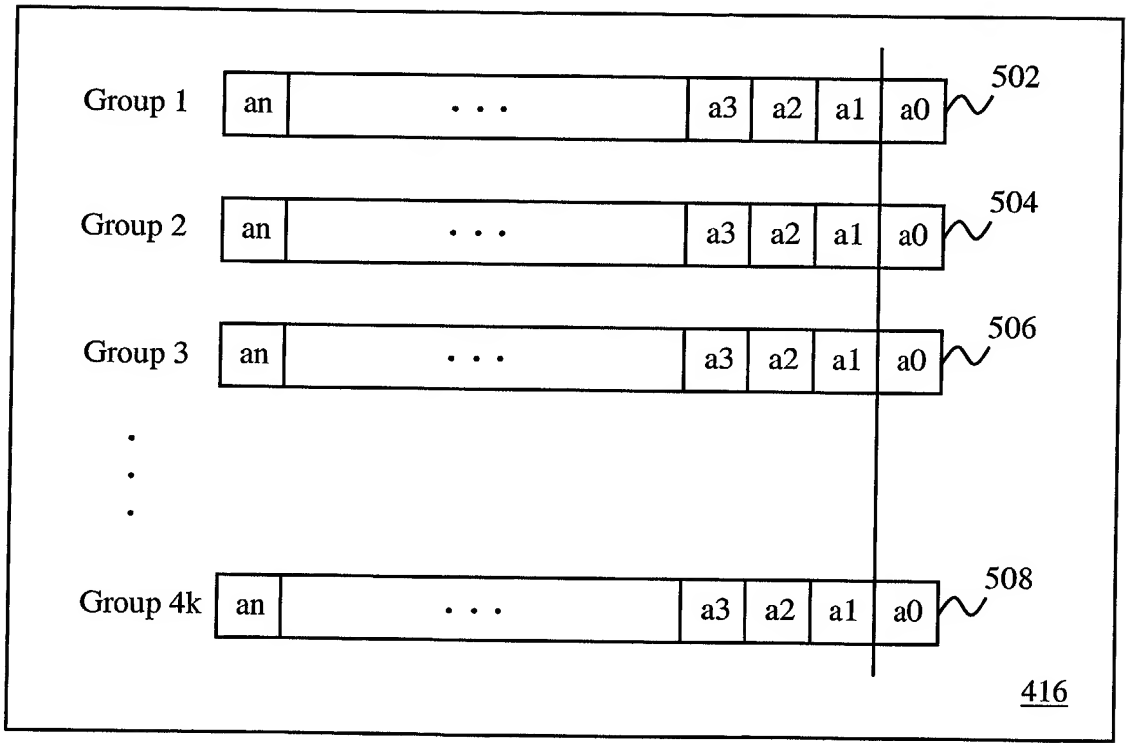
Group 1
an
. . .
a3
a2
a1
a0
502
Group 2
an
. . .
a3
a2
a1
a0
504
Group 3
an
. . .
a3
a2
a1
a0
506
Group 4k
an
. . .
a3
a2
a1
a0
508
416

Figure 5

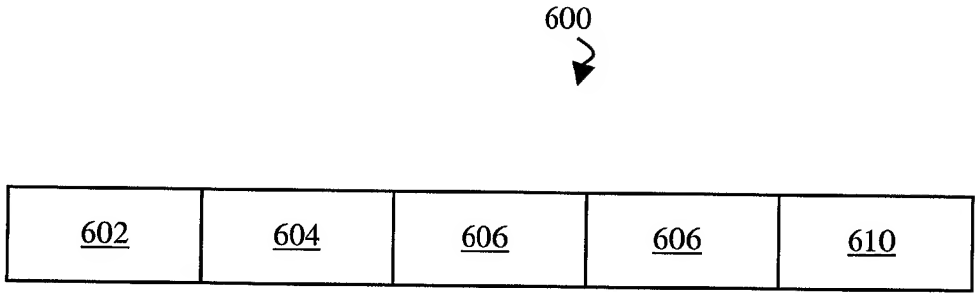

Figure 6

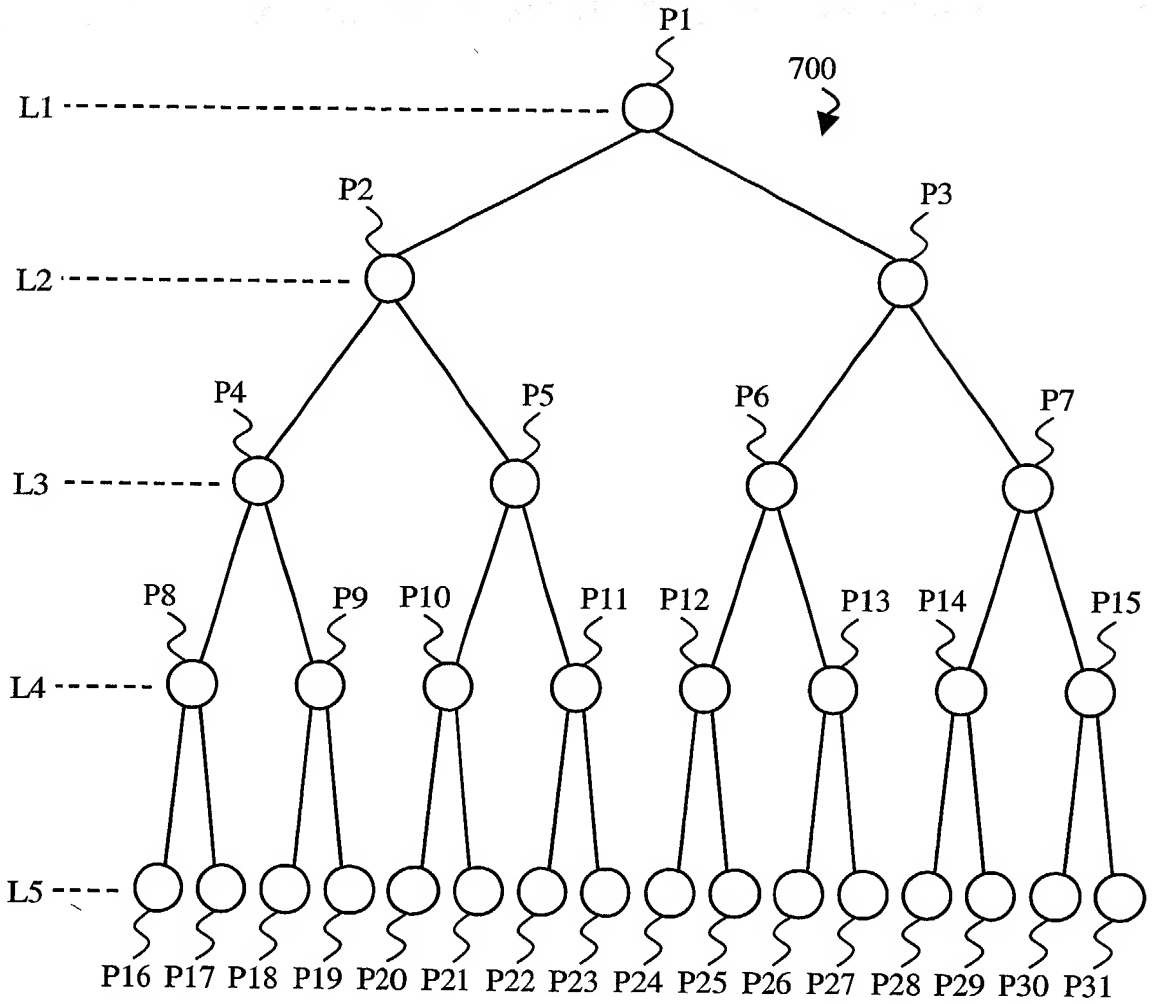
P1
700
L1
P2
P3
L2
P4
P5
P6
P7
L3
P8
P9
P10
P11
P12
P13
P14
P15
L4
L5
P16 P17 P18 P19 P20 P21 P22 P23 P24 P25 P26 P27 P28 P29 P30 P31

Figure 7

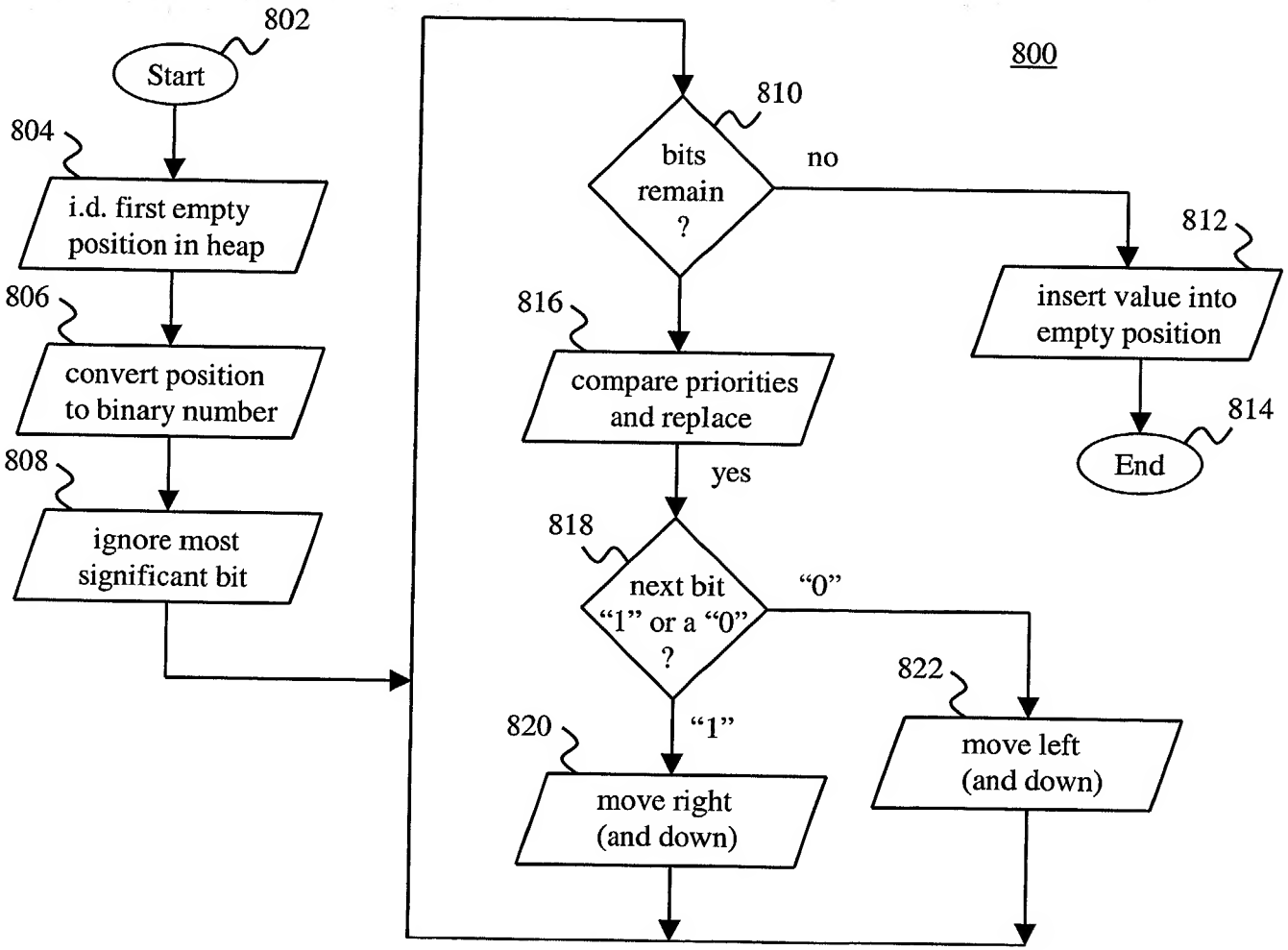
800
802
Start
804
i.d. first empty position in heap
806
convert position to binary number
808
ignore most significant bit
810
bits remain ?
no
812
insert value into empty position
814
End
816
compare priorities and replace
yes
818
next bit "1" or a "0" ?
"0"
"1"
820
move right (and down)
822
move left (and down)

Figure 8

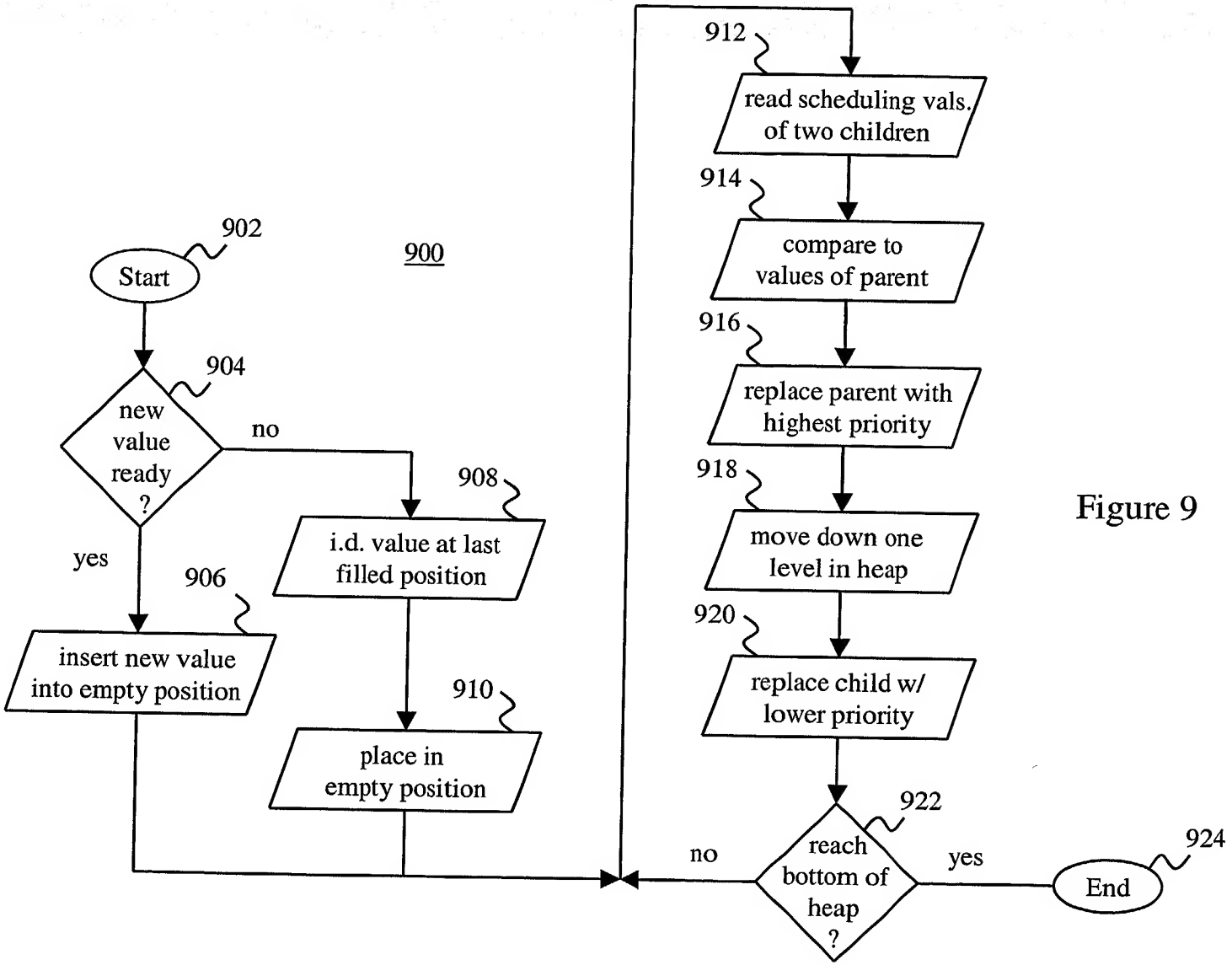
900
902
Start
904
new value ready ?
no
yes
906
insert new value into empty position
908
i.d. value at last filled position
910
place in empty position
912
read scheduling vals. of two children
914
compare to values of parent
916
replace parent with highest priority
918
move down one level in heap
920
replace child w/ lower priority
922
reach bottom of heap ?
no
yes
924
End

Figure 9

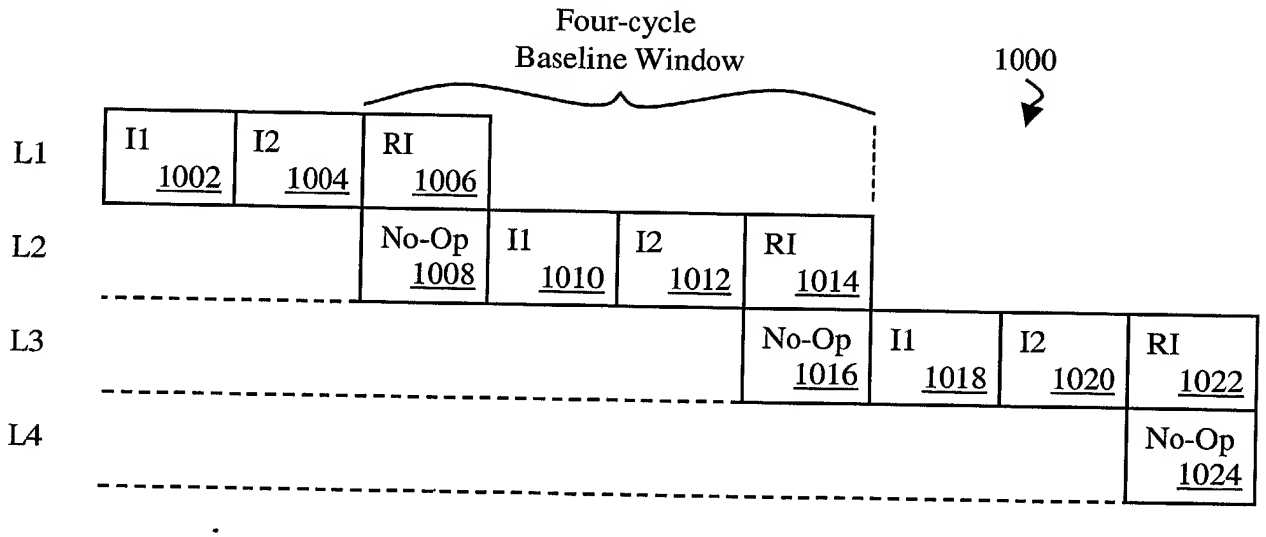
Four-cycle
Baseline Window
1000
L1
L2
L3
L4
I1 1002
I2 1004
RI 1006
No-Op 1008
I1 1010
I2 1012
RI 1014
No-Op 1016
I1 1018
I2 1020
RI 1022
No-Op 1024

Figure 10

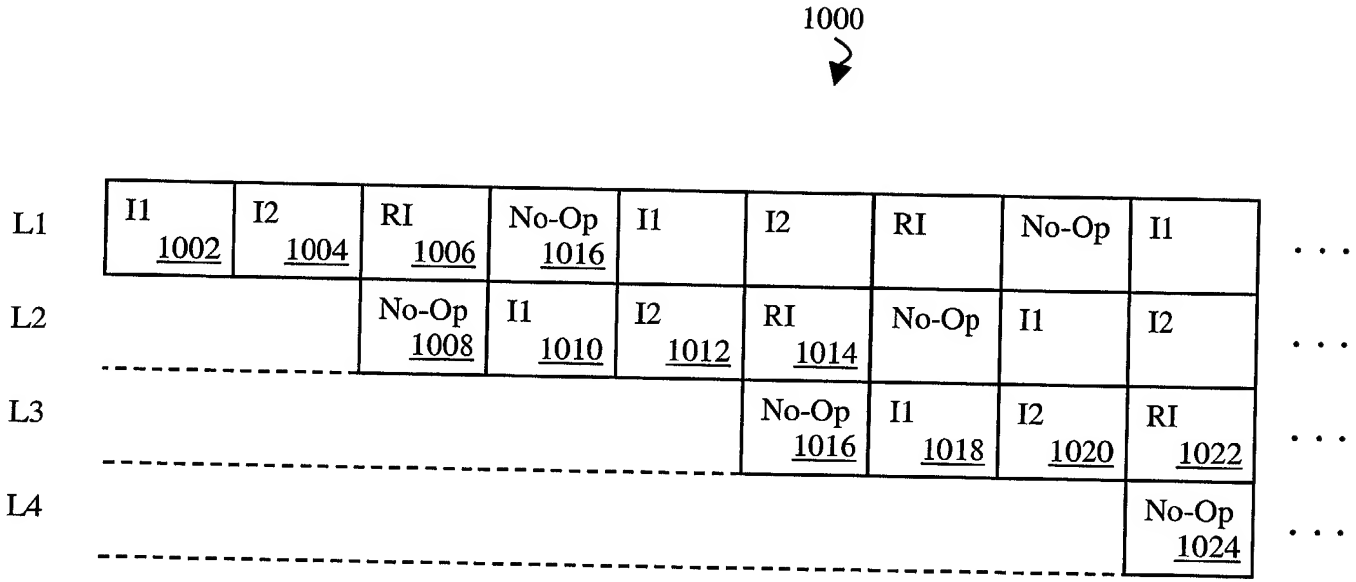
1000
L1
L2
L3
L4
I1 1002
I2 1004
RI 1006
No-Op 1016
I1
I2
RI
No-Op
I1
No-Op 1008
I1 1010
I2 1012
RI 1014
No-Op
I1
I2
No-Op 1016
I1 1018
I2 1020
RI 1022
No-Op 1024

Figure 11

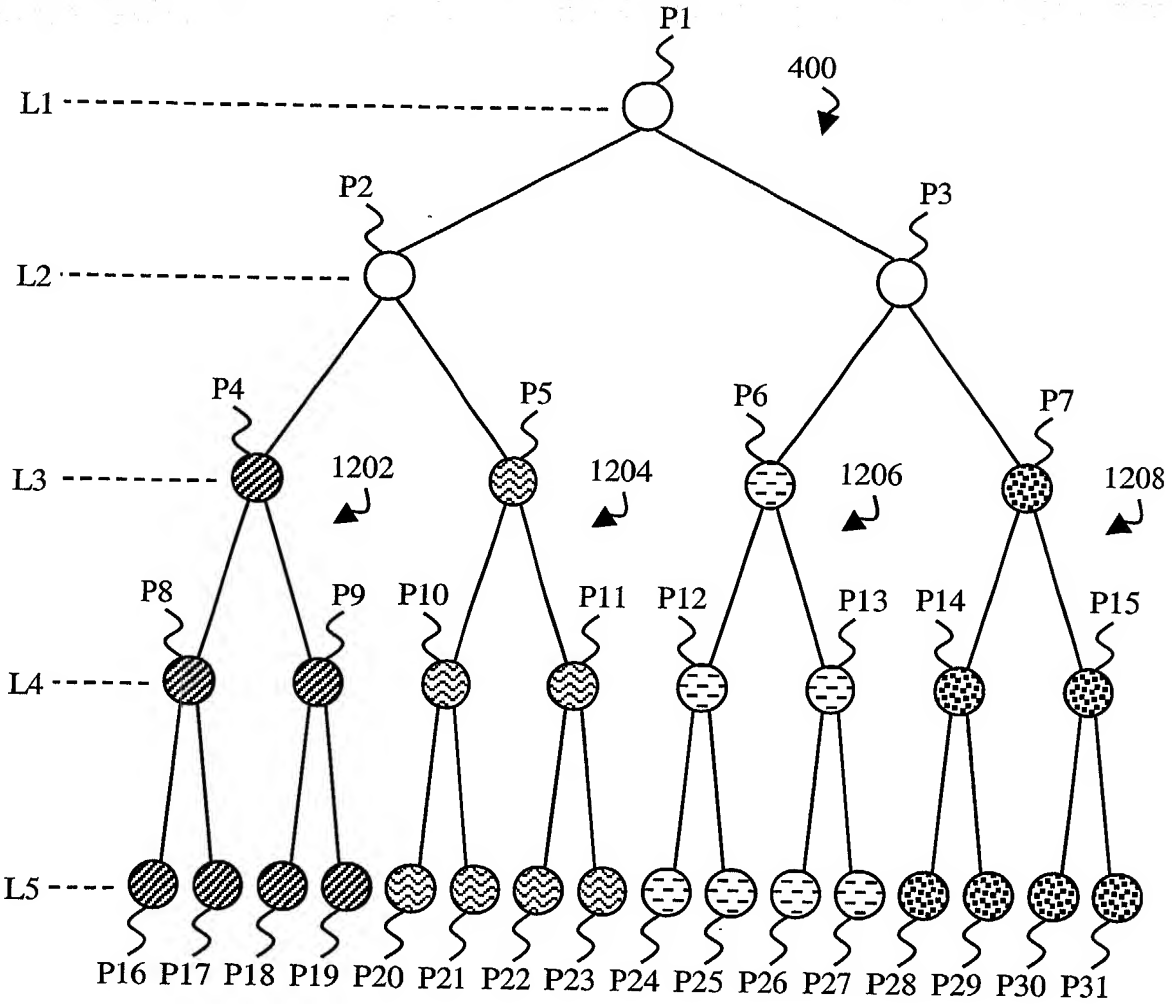
P1
400
L1
P2
P3
L2
P4
P5
P6
P7
L3
1202
1204
1206
1208
P8
P9
P10
P11
P12
P13
P14
P15
L4
L5
P16 P17 P18 P19 P20 P21 P22 P23 P24 P25 P26 P27 P28 P29 P30 P31

Figure 12

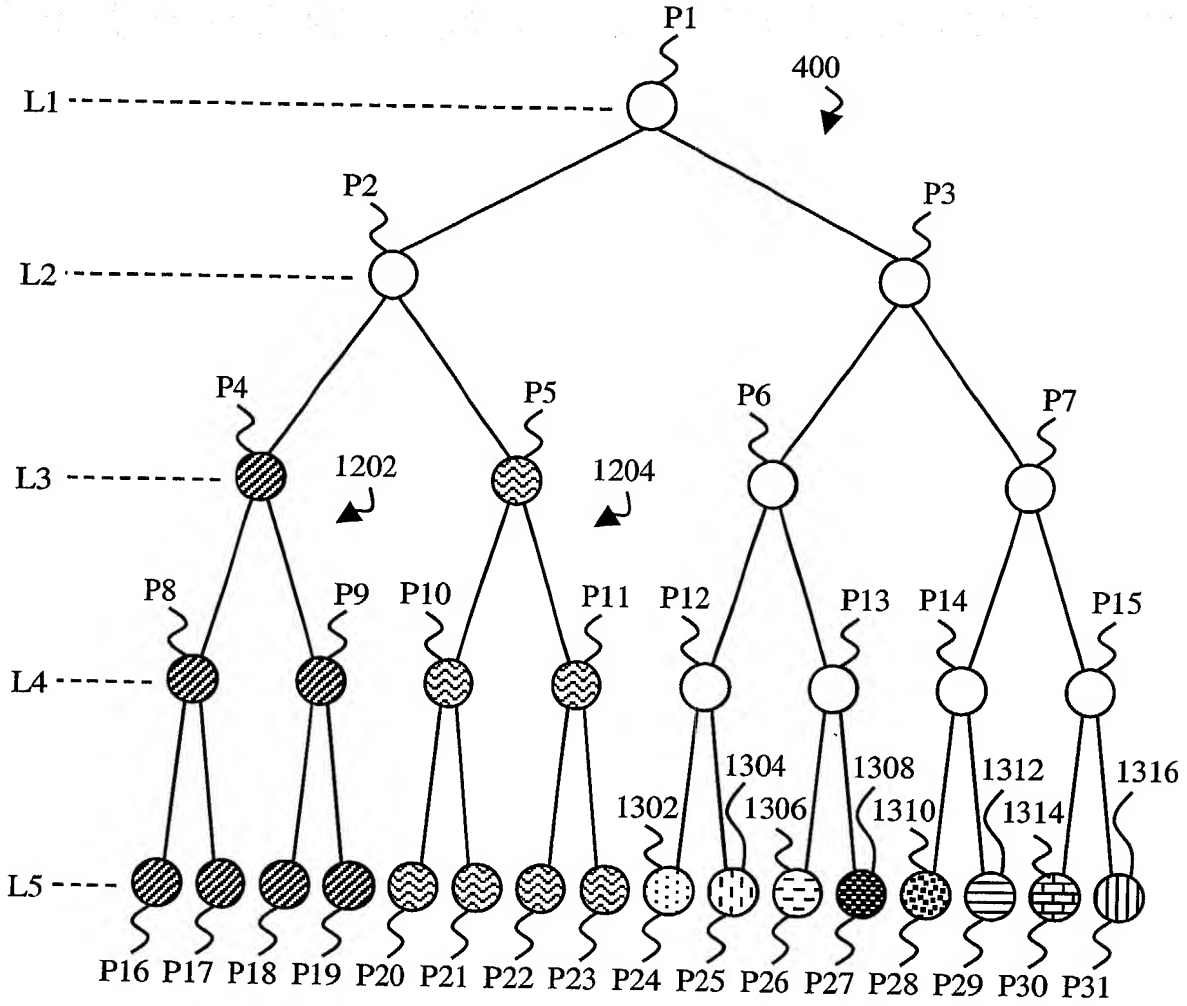
P1
400
L1
P2
P3
L2
P4
P5
P6
P7
L3
1202
1204
P8
P9
P10
P11
P12
P13
P14
P15
L4
1304
1308
1312
1316
1302
1306
1310
1314
L5
P16 P17 P18 P19 P20 P21 P22 P23 P24 P25 P26 P27 P28 P29 P30 P31

Figure 13

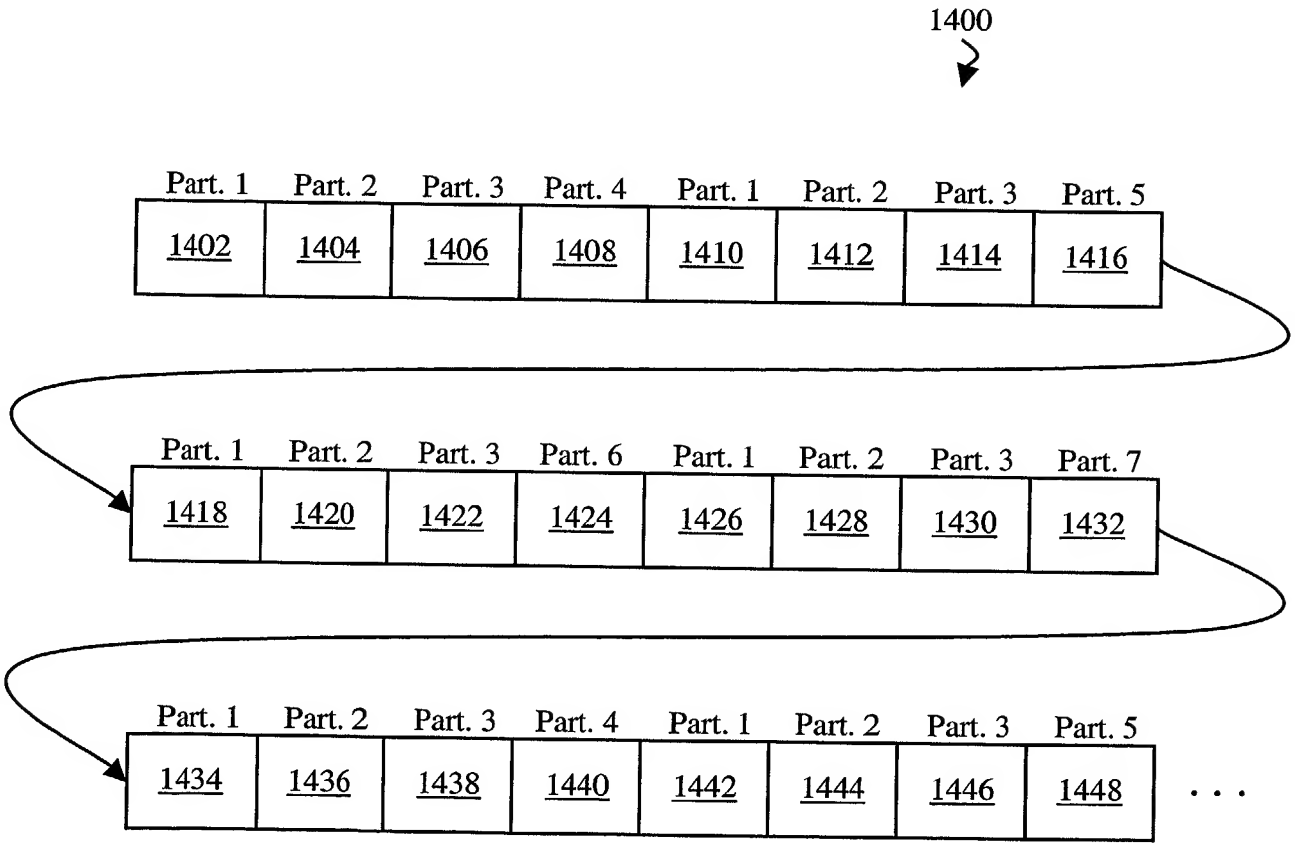
1400
Part. 1
Part. 2
Part. 3
Part. 4
Part. 1
Part. 2
Part. 3
Part. 5
1402
1404
1406
1408
1410
1412
1414
1416
Part. 1
Part. 2
Part. 3
Part. 6
Part. 1
Part. 2
Part. 3
Part. 7
1418
1420
1422
1424
1426
1428
1430
1432
Part. 1
Part. 2
Part. 3
Part. 4
Part. 1
Part. 2
Part. 3
Part. 5
1434
1436
1438
1440
1442
1444
1446
1448
. . .

Figure 14

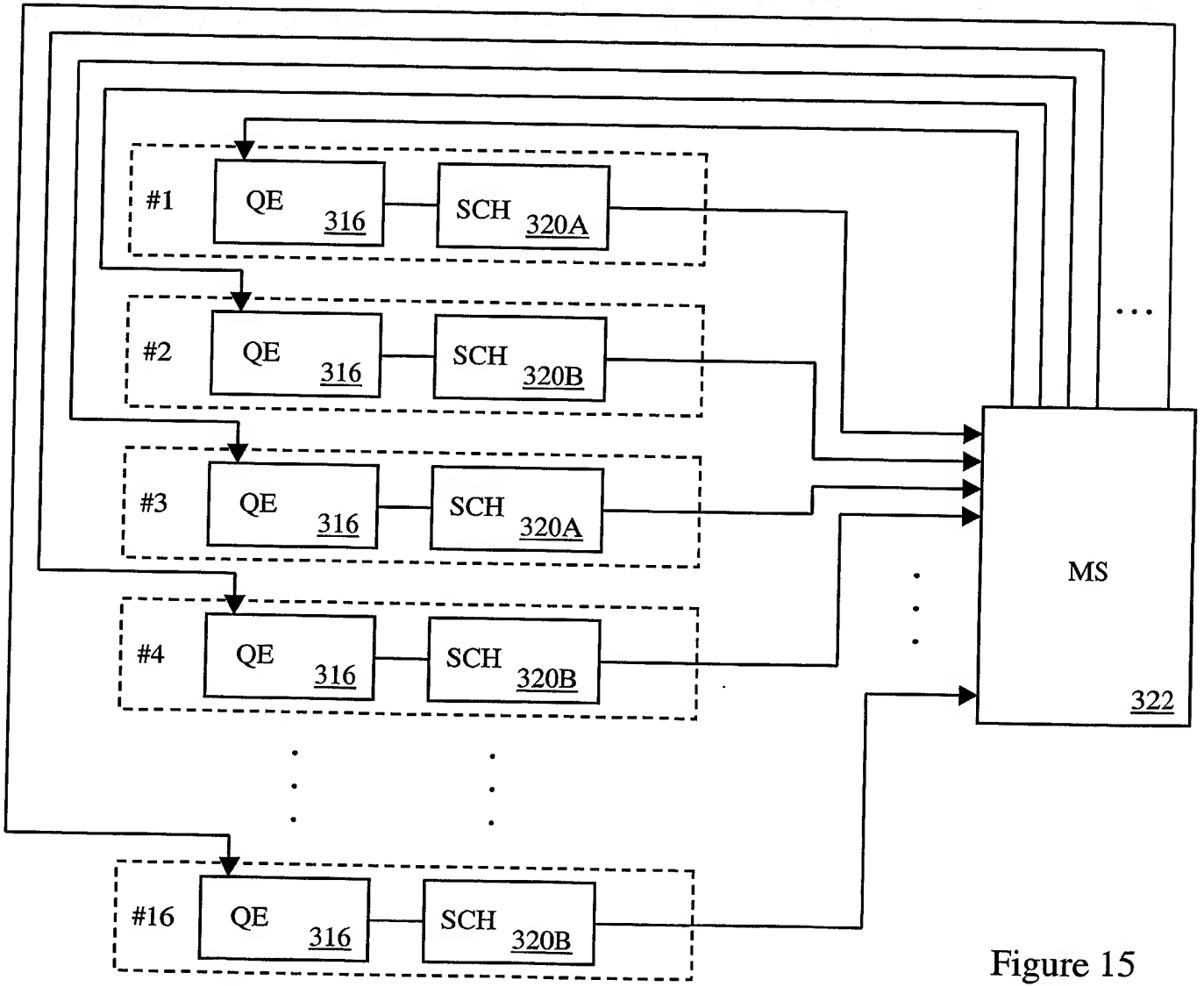
#1
QE
316
SCH
320A
#2
QE
316
SCH
320B
#3
QE
316
SCH
320A
#4
QE
316
SCH
320B
#16
QE
316
SCH
320B
MS
322

Figure 15

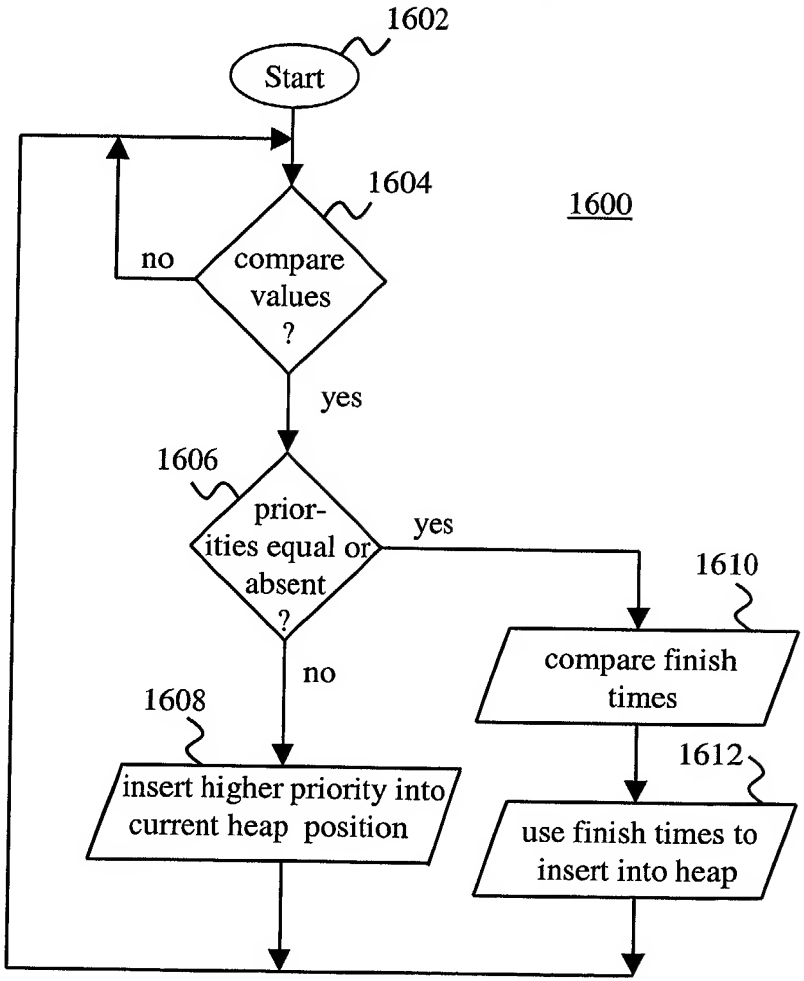
1602
Start
1604
1600
compare values ?
no
yes
1606
prior-ities equal or absent ?
yes
1610
compare finish times
no
1608
insert higher priority into current heap position
1612
use finish times to insert into heap

Figure 16

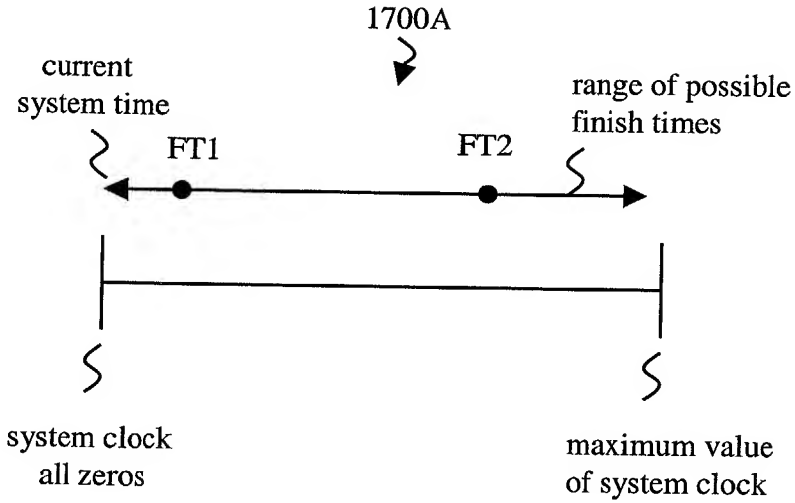

Figure 17A

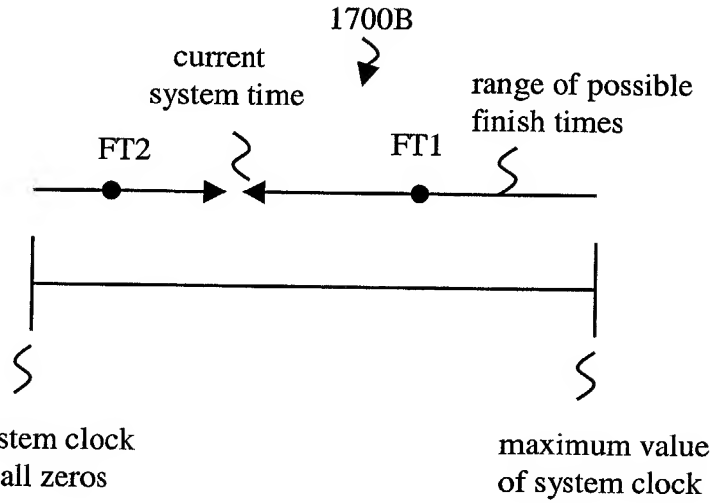

Figure 17B

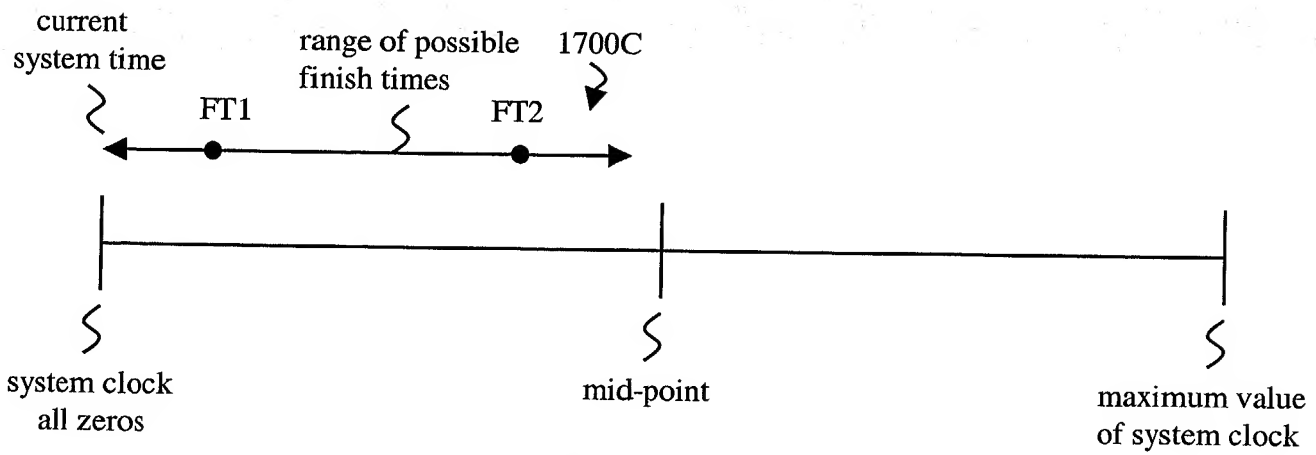

Figure 17C

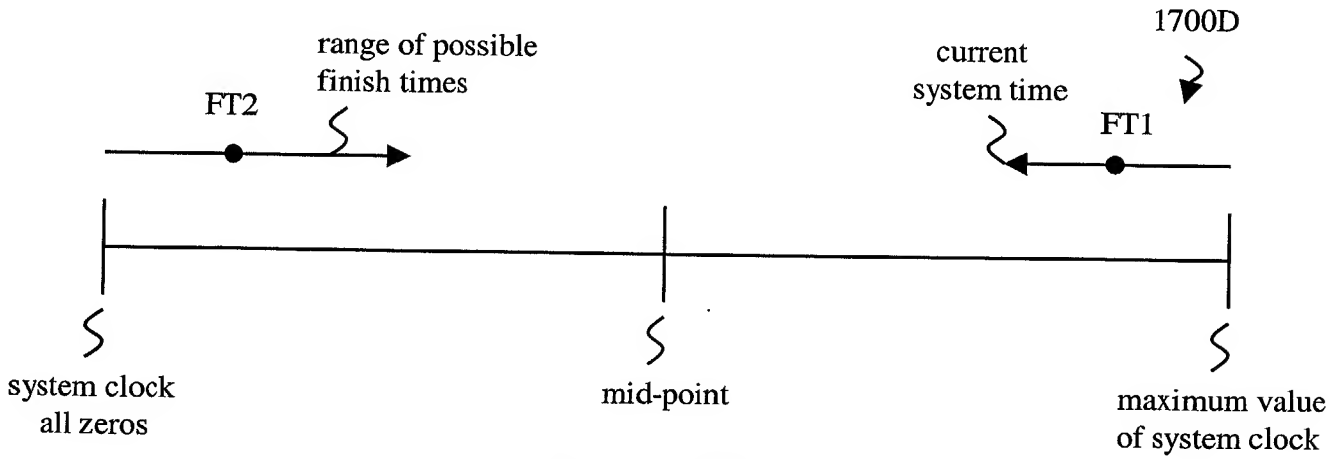

Figure 17D

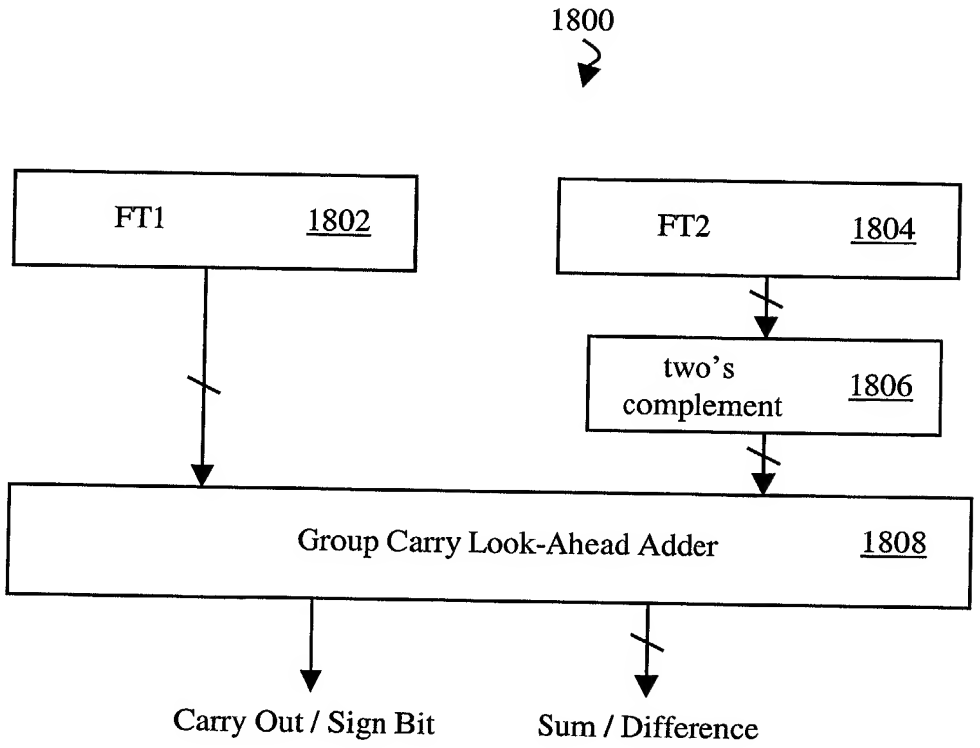
1800
FT1
1802
FT2
1804
two's complement
1806
Group Carry Look-Ahead Adder
1808
Carry Out / Sign Bit
Sum / Difference

Figure 18

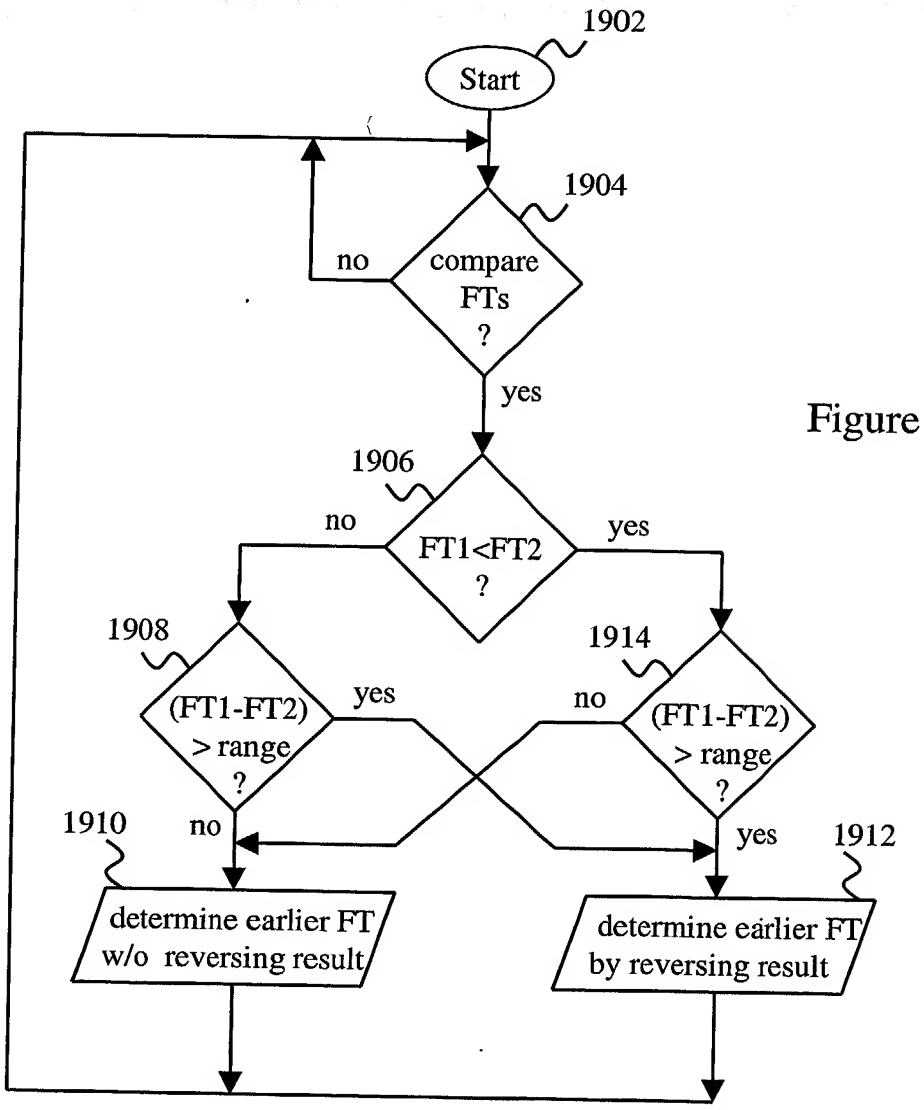
1902
Start
1900
1904
compare FTs ?
no
yes
1906
FT1<FT2 ?
no
yes
1908
(FT1-FT2) > range ?
yes
no
1914
(FT1-FT2) > range ?
no
yes
1910
determine earlier FT w/o reversing result
1912
determine earlier FT by reversing result

Figure 19